



# 物联网中的分布式数据感知 与并行处理算法设计

殷亚凤 助理研究员

南京大学 计算机科学与技术系



# 提纲

---

**1**

**背景介绍**

**2**

**分布式数据感知**

**3**

**并行处理算法设计**

**4**

**总结**

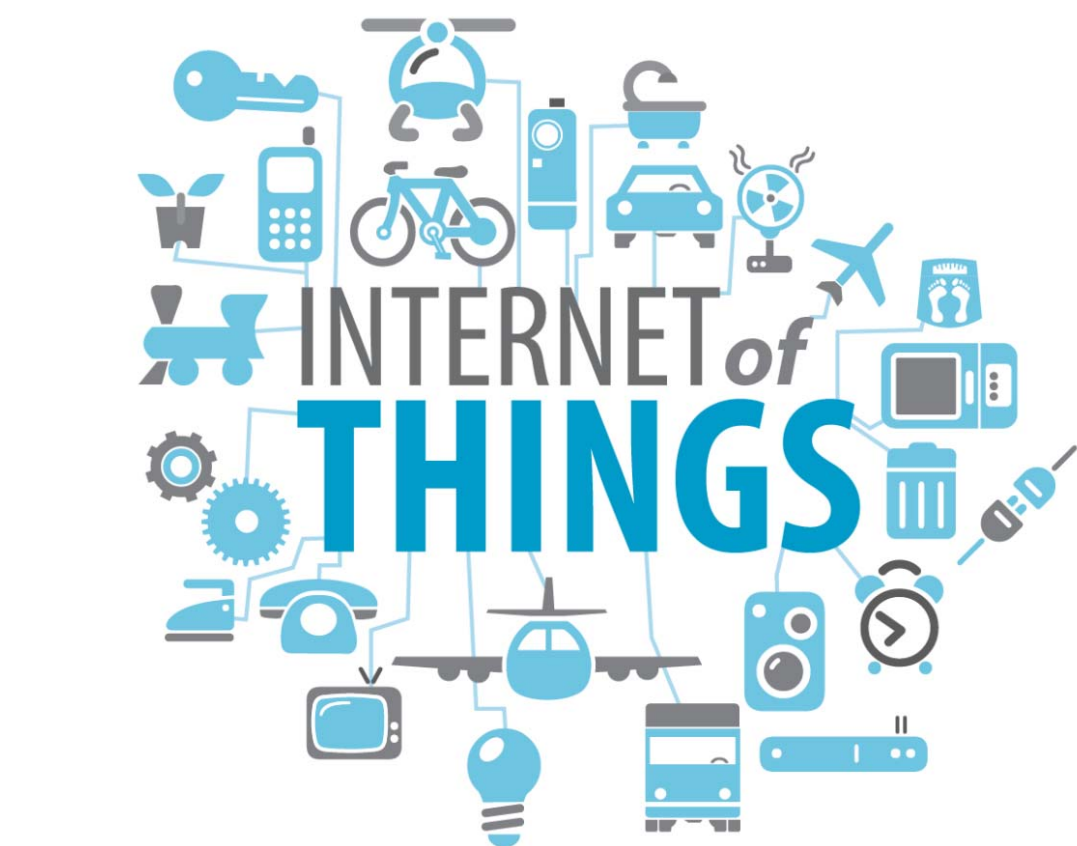
# 背景介绍

## • 物联网的起源



## 15.3.1 15.3.1 15.3.1

## RESEARCH DESIGN

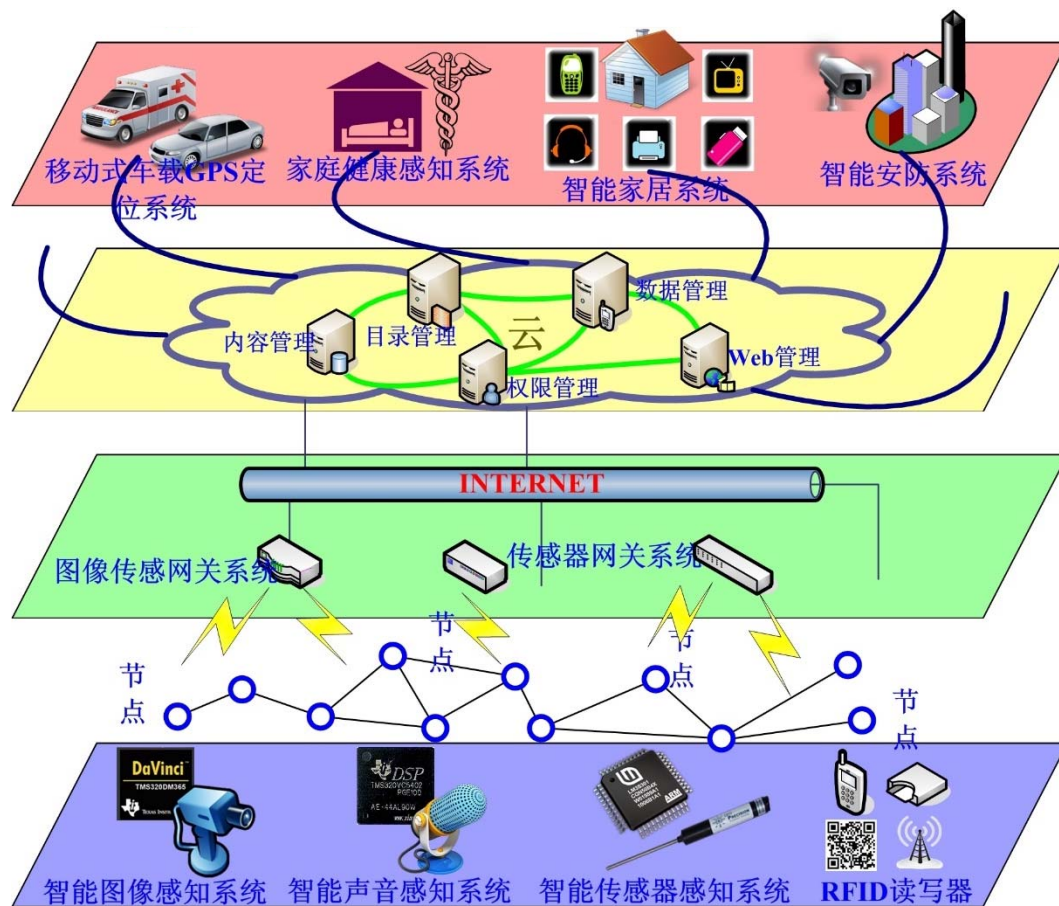


物联网是一个基于互联网、传统电信网等信息载体，让所有能够被独立寻址的**普通物理对象实现互联互通**的网络。

- 普通对象设备化
- 自治终端互联化
- 普适服务智能化

# 背景介绍

## • 物联网四层模型



**综合应用层:** 从以数据服务为主要特征的文件传输到以用户为中心的应用

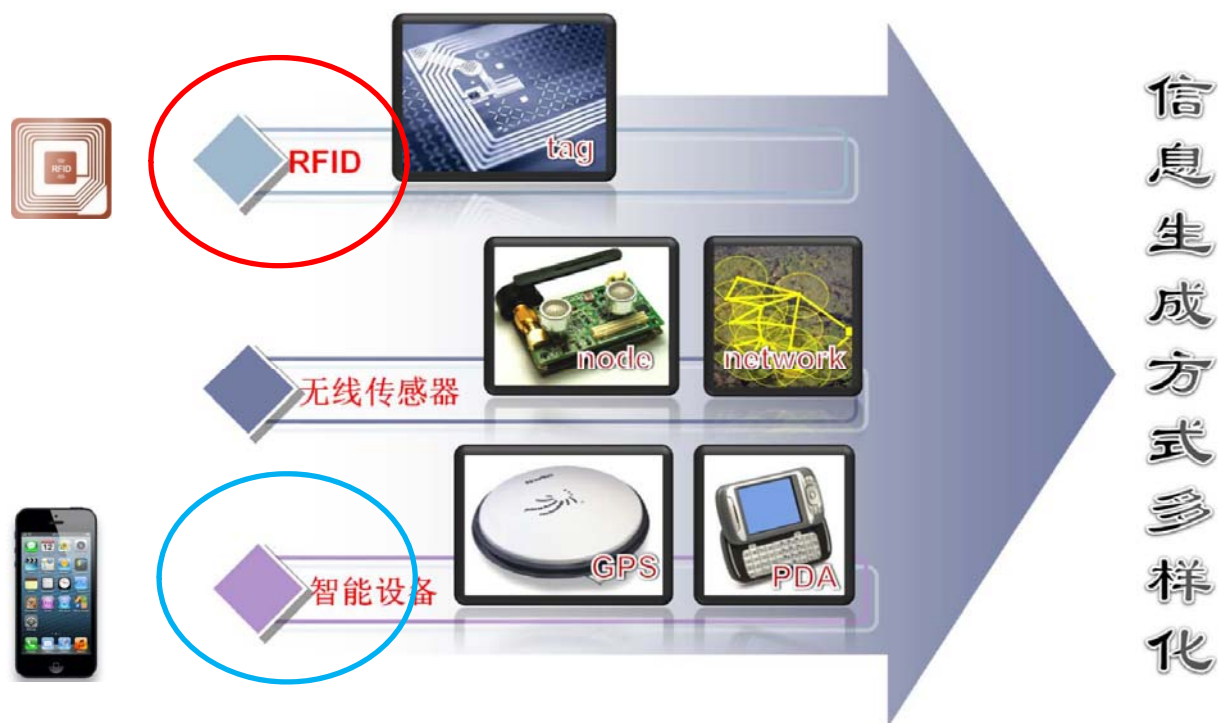
**管理服务层:** 将大规模数据组织起来，为上层应用提供支撑。

**网络构建层:** 把下层设备接入互联网，供上层服务使用，包括3G、4G、WiFi、蓝牙、NFC等通信技术

**感知识别层:** 联系物理世界和信息世界，包括RFID、无线传感器、移动终端等感知设备

# 背景介绍

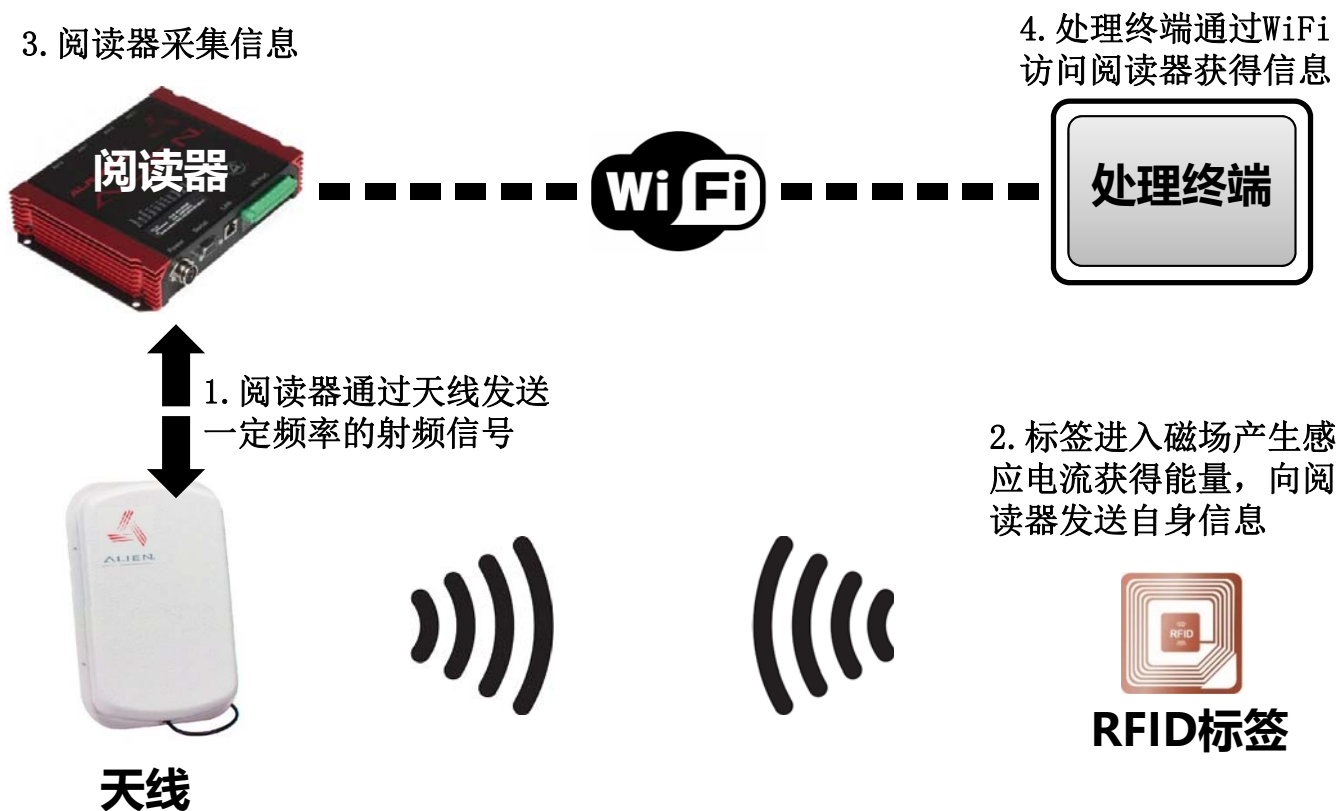
- 感知识别层——核心技术



通过感知识别技术，让物品“开口说话、发布信息”，是融合物理世界和信息世界的重要一环，是物联网区别于其他网络的**最独特**的部分。

# 数据感知

## 感知识别层——射频识别技术

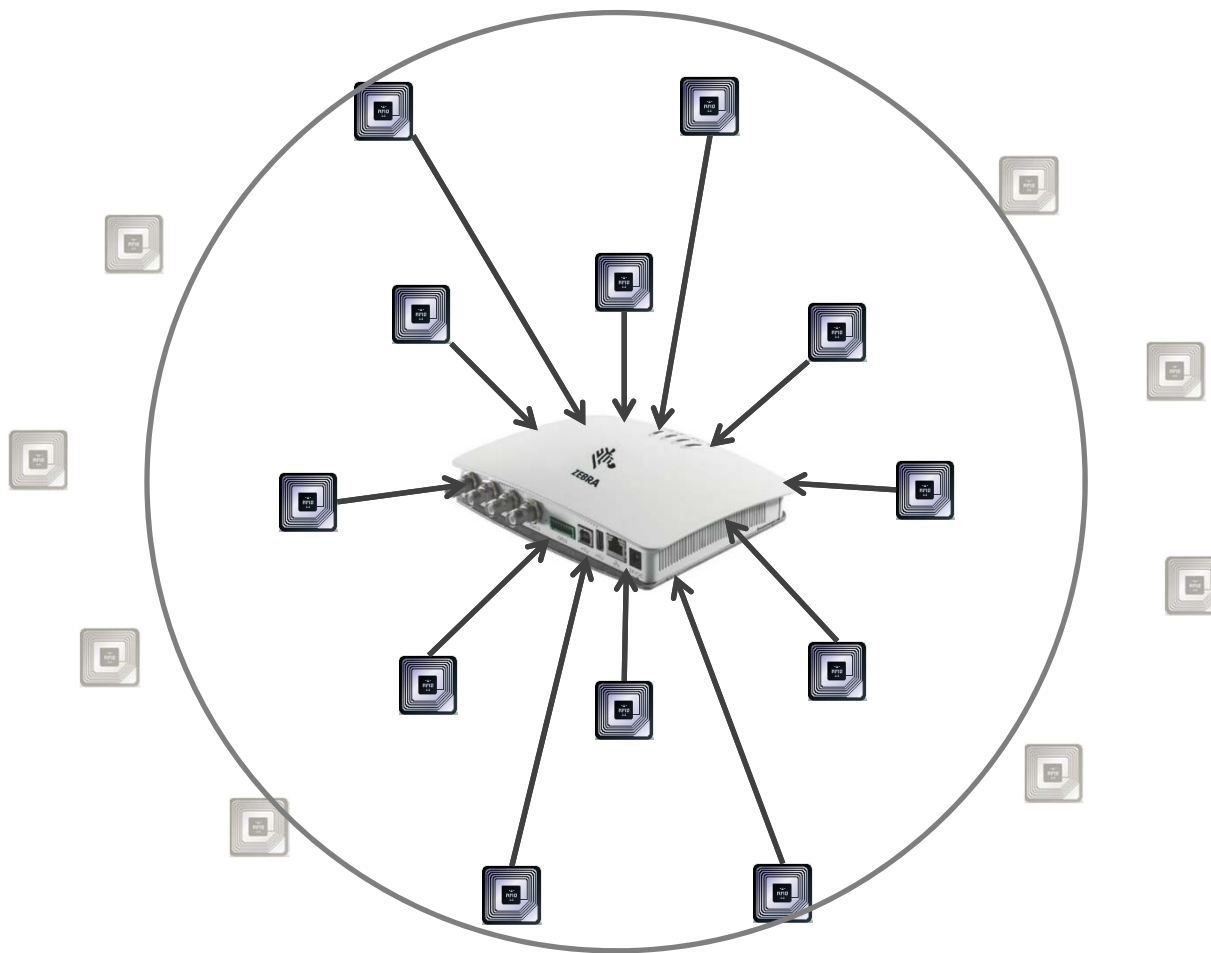




# 射频识别技术

- 单一阅读器的RFID系统

集中式

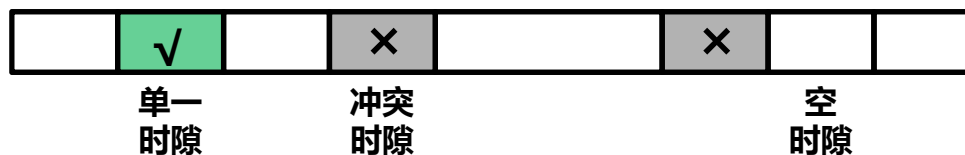
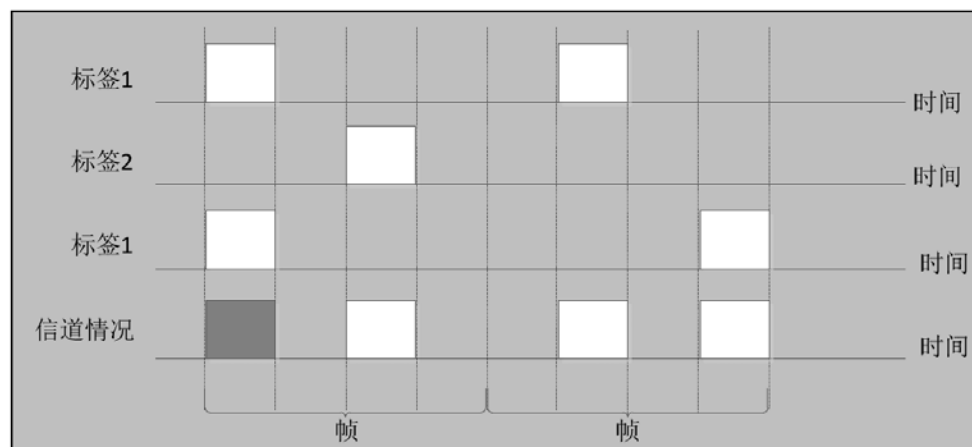




# 射频识别技术

## RFID中的标签识别算法

### 基于帧的分时隙ALOHA防冲突算法



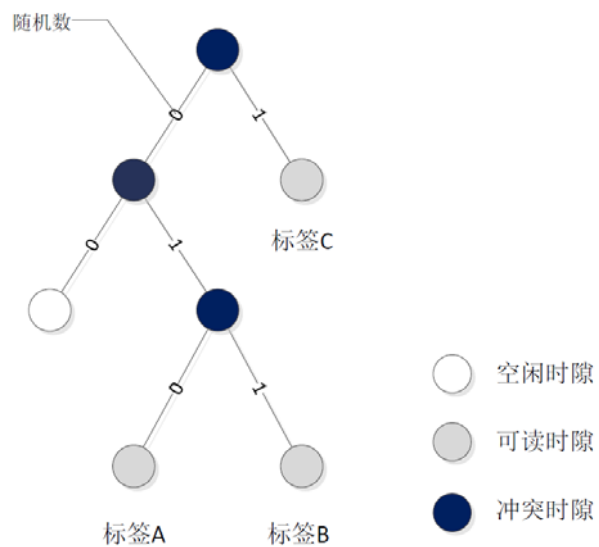
识别过程：

- 阅读器广播帧的长度 $f$ ，即时隙个数；
- 标签接收到帧长后随机独立地在 $0-(f-1)$ 中选择一个时隙序号，记在SN中；
- 阅读器启动一个新的时隙，如果标签SN为零则发送标识符，反之SN减1；
- 如果时隙无冲突，该标签进入休眠；
- 如果时隙有冲突，标签进入等待，重新选择时隙发送；

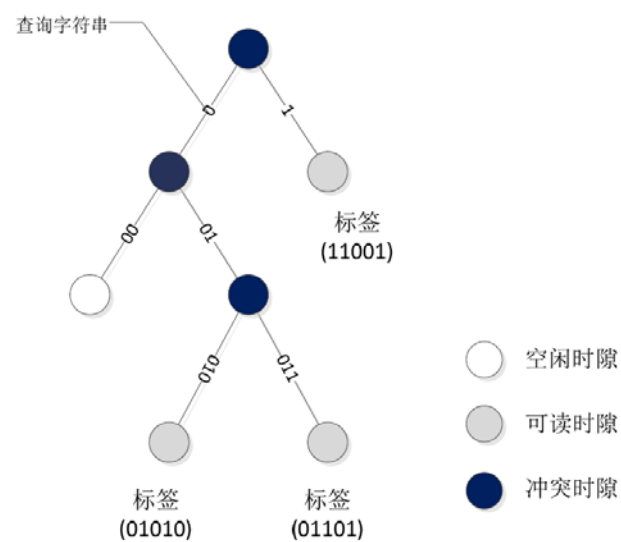
# 射频识别技术

- RFID中的标签识别算法

## 基于二进制树的防冲突算法



随机二进制树

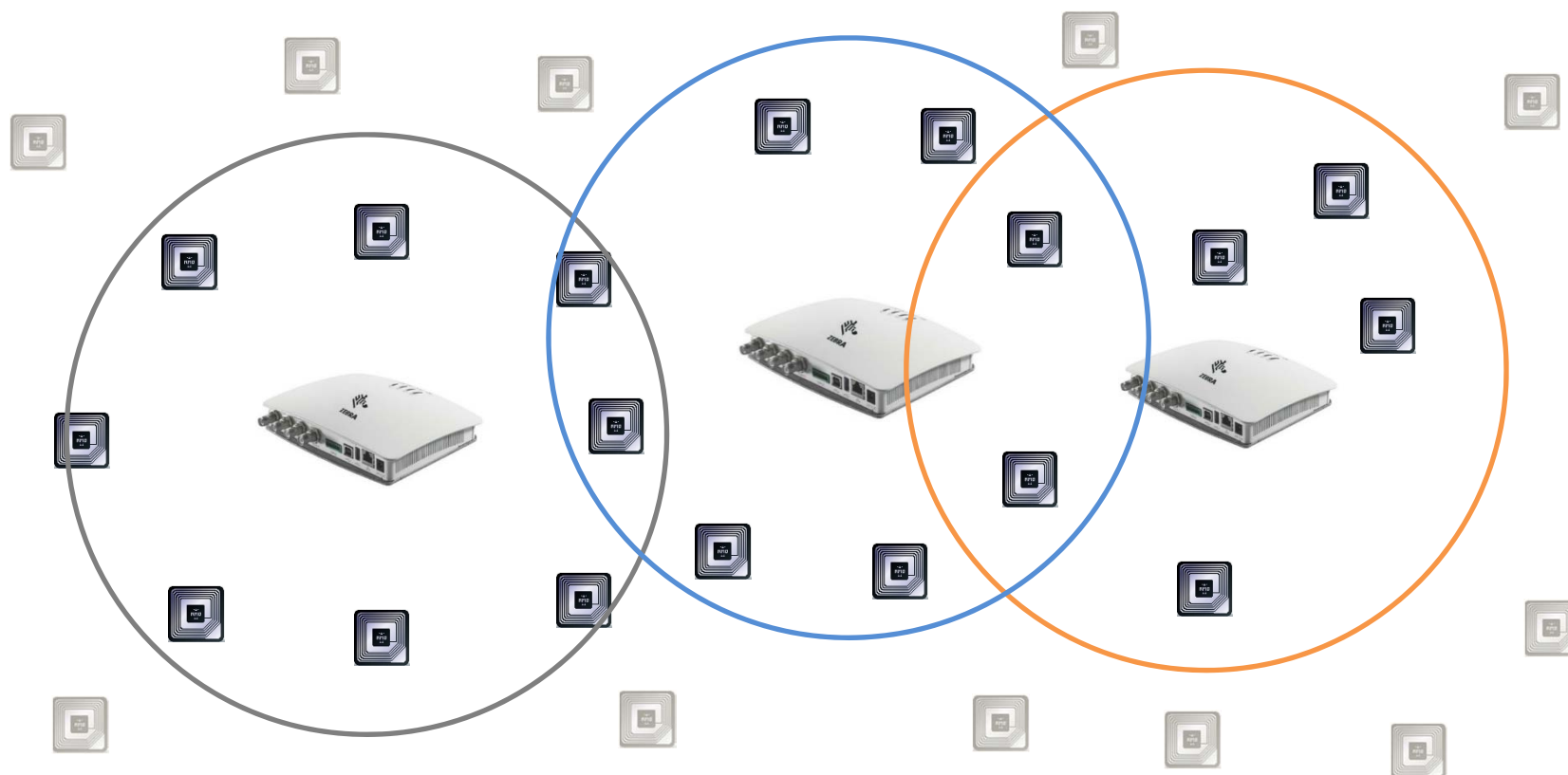


查询二进制树

# 分布式数据感知

- 大规模的多阅读器RFID系统

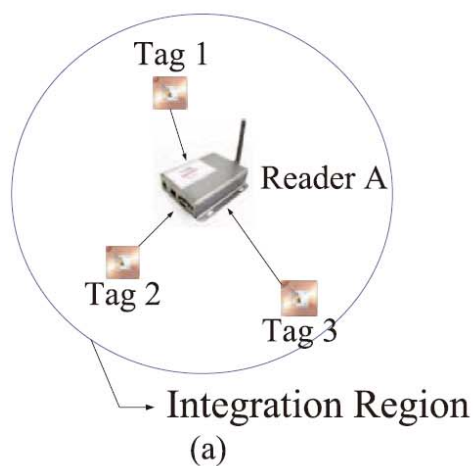
分布式



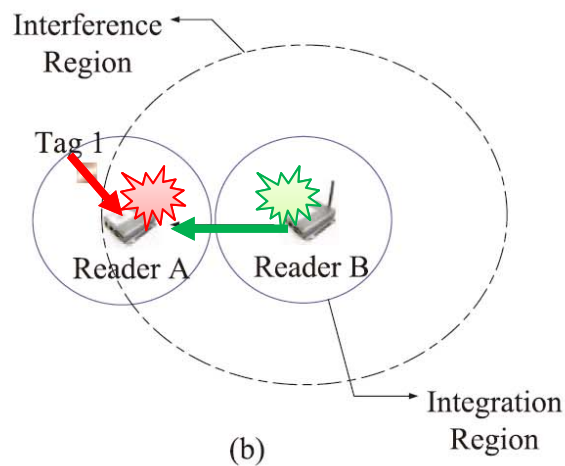
# 分布式数据感知

- 大规模的多阅读器RFID系统

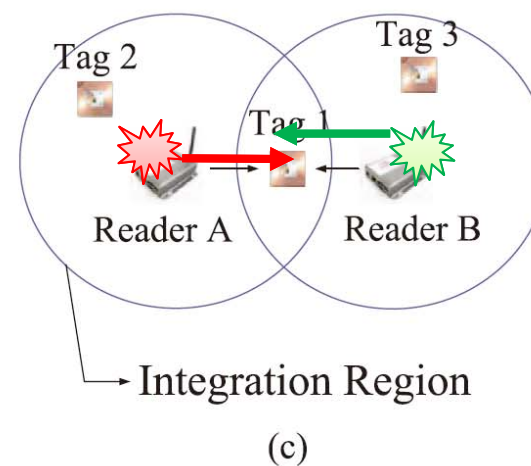
Tang, ShaoJie, et al. "Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems." ICNP 2009.



标签-标签冲突



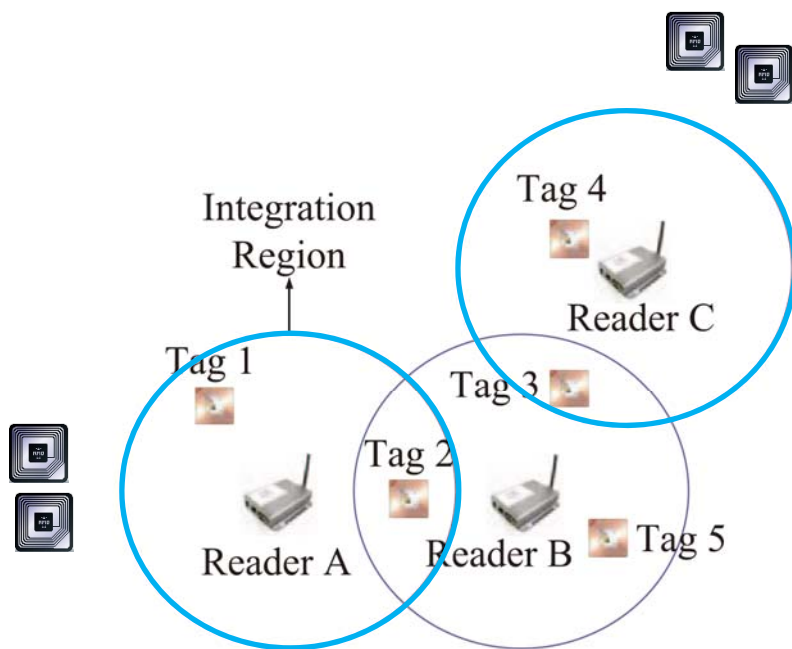
阅读器-标签冲突



阅读器-阅读器冲突

# 分布式数据感知

## 大规模的多阅读器RFID系统



问题描述：

阅读器

$n$  readers  $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$

$I_t = \{r_i : r_i \text{ is activated at time slot } t\}$

标签

$A_t(i)$ :  $t$ 时刻, 出现在 $r_i$ 扫描范围内的**新标签**

$X_t = \{X_t(i) : i = 1, \dots, n\}$ :  $t$ 时刻, 阅读器 $r_i$ 扫描范围内未被识别标签

$\omega(I_t, t)$ :  $t$ 时刻, 可被识别的标签总数

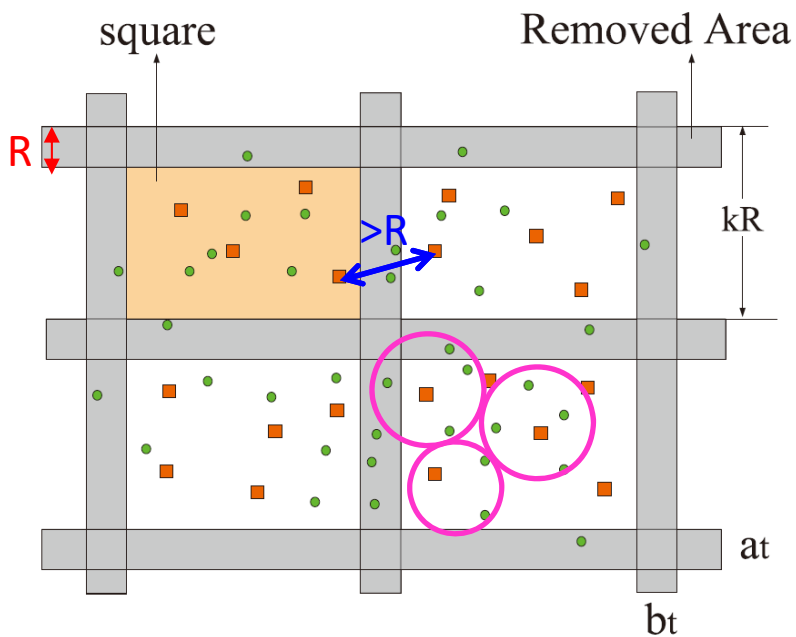
如何**选择更少的阅读器**读取更多的标签?  
(避免阅读器-阅读器冲突, 阅读器-标签冲突)

Maximum Weighted Independent Set (MWIS)

## 第 1 章 绪论

\_\_\_\_\_

Estimate **global spatial distribution** of tags

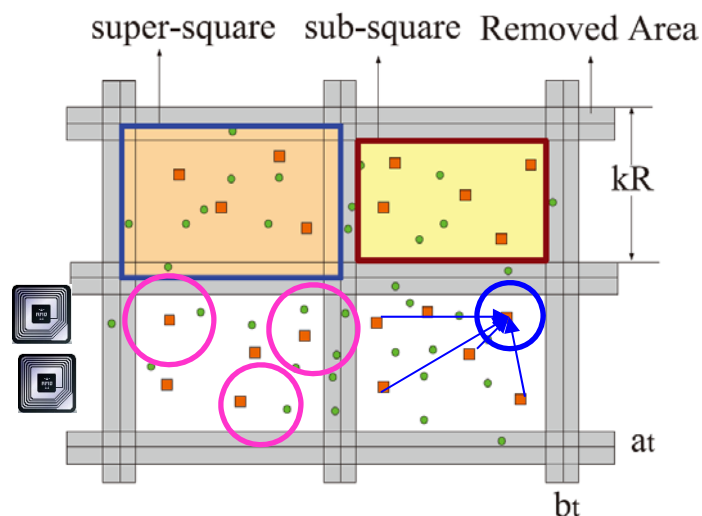


(a) centralized method

- 将二维部署空间**划分为多个grids**；
- **灰色区域的阅读器不被激活**；  
( $R > 2r$ ,  $r$ 是阅读器扫描范围)
- 一个**Square中的阅读器求MWIS**；  
(**枚举**)
- **综合每个Square的MWIS结果**，获得整个区域的MWIS结果（即选择需要被激活的阅读器）；

# 分布式数据感知

## 分布式阅读器激活方案



(b) distributed method

This set can clearly be computed locally as follows: If  $\omega(I_{t-1}^{(i,j)}, t) > \omega(I_t^{sub(i,j)}, t)$ , let  $I_t^{(i,j)} = I_{t-1}^{(i,j)}$ , otherwise  $I_t^{(i,j)} = I_t^{sub(i,j)}$ . Then we use  $\bigcup_{i,j} I_t^{(i,j)}$  for all super-squares as a global solution.

Only estimate its **local spatial distribution** of tags

- 在分布式方法中，如何通过**局部**的解决方案获得**全局**的解决方案？
- 局部解决方案更新时，如何更新全局解决方案？



- 在Super-square中选择则一个阅读器作为唯一的**协调者**（激活的阅读器）；
- 根据协调者**估算super-square中的标签分布**，**选取Sub-square中的其他阅读器**；
- 将**t时刻Sub-square**中选取的方案与**t-1时刻Super-square**中的方案进行**比较**，选择**最优方案**并更新给Super-square；
- 综合各个Super-square的方案，作为最终结果。



# 分布式数据感知

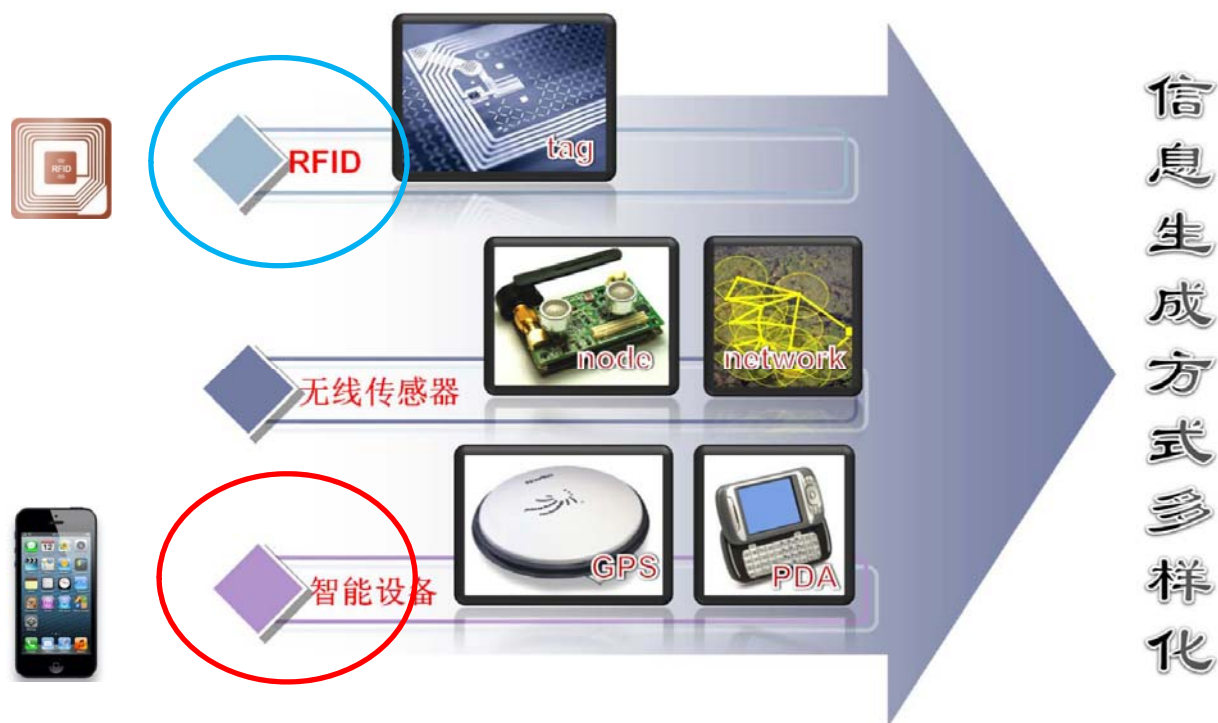
- 大规模场景下的集中式数据感知与分布式数据感知

	比较角度	集中式	分布式
1	全局信息 ( 如标签分布情况 )	已知 ( 如事先估算全局信息 )	未知
2	计算复杂度	大	小
3	计算最优方案	近似算法	近似算法
4	大规模数据收集方式	分治思想	分治思想
5	子区域数据收集方式	根据已知的全局/局部 信息，计算局部最优	协作计算、迭代优化

其中，比较角度3~5主要针对大规模多阅读器的RFID系统得出的分析结论。

# 移动设备

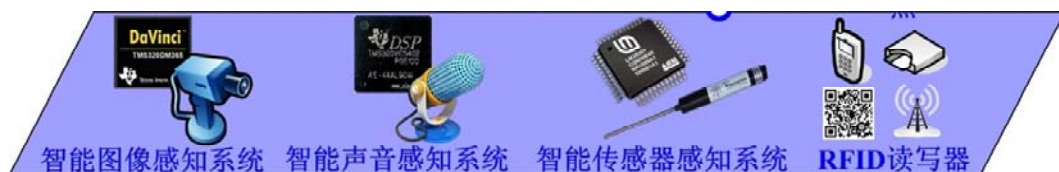
- 感知识别层——核心技术



通过感知识别技术，让物品“开口说话、发布信息”，是融合物理世界和信息世界的重要一环，是物联网区别于其他网络的**最独特**的部分。

# 移动设备

- 感知识别层——移动设备



**感知识别层: 联系物理世界和信息世界, 包括RFID、无线传感器、移动终端等感知设备**

移动设备是一个**小型**（口袋大小）的**计算设备**，装有一个**操作系统**，可以通过WiFi、蓝牙、近场通信等方式接入互联网或其他设备。



平板电脑



个人数字助理



智能手机



个人导航设备



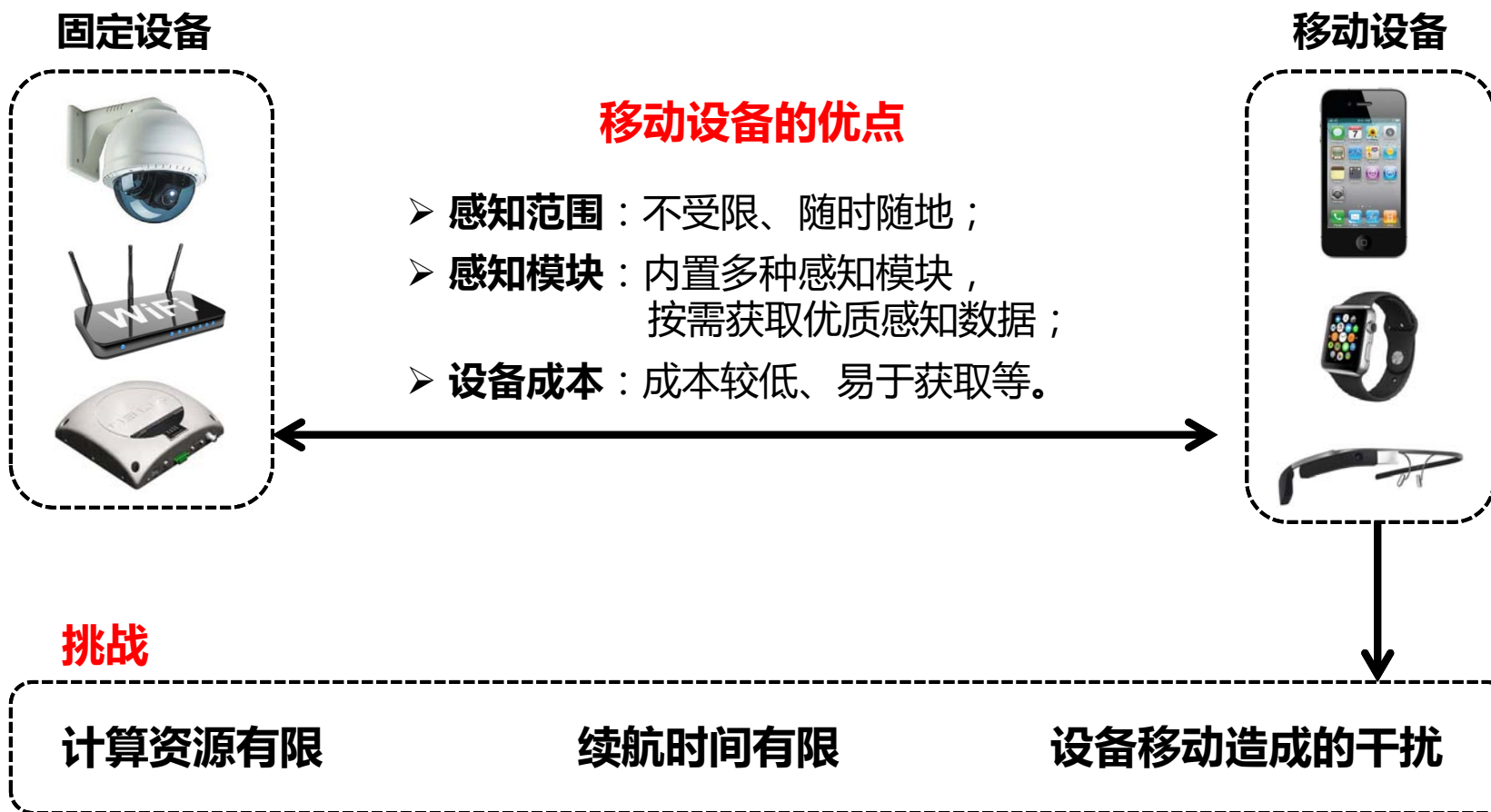
智能手表



头戴式显示器

# 移动设备

## • 移动设备的特点

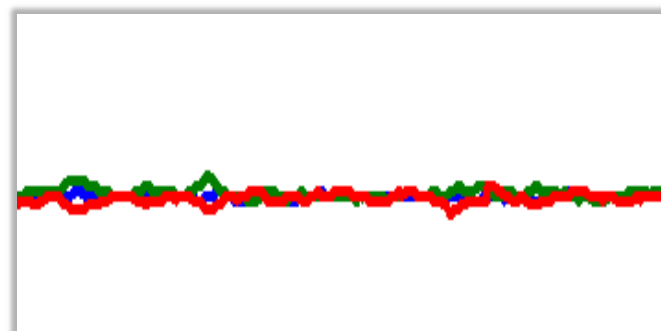


# 移动设备

- 微动作感知识别



细粒度动作



# 微动作感知识别

## • 现有解决方案（按键操作为例）



### FingerRing:

利用**指环**上的加速度计检测手指运动  
(M. Fukumoto et al., CHI'97)



### UbiK:

利用麦克风实现按键操作定位（需要**指甲**配合按键操作）  
(J. Wang et al., MobiSys'14)



使用可见光/红外投影键盘

(<http://www.ctxtechnologies.com/products/vk-200-keyfob-virtual-keyboard/>)

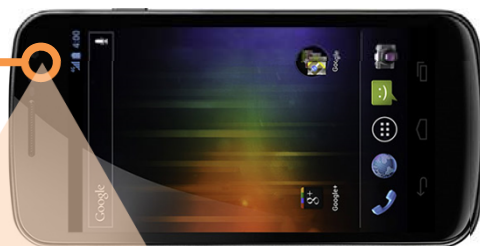
## 缺点

- 引入额外的设备负担，
- 改变用户打字习惯，
- 用户体验较差等。

# 基于目标定位的微动作感知识别机制

## • 问题描述

内置摄像头



移动设备  
(智能手机)

纸键盘

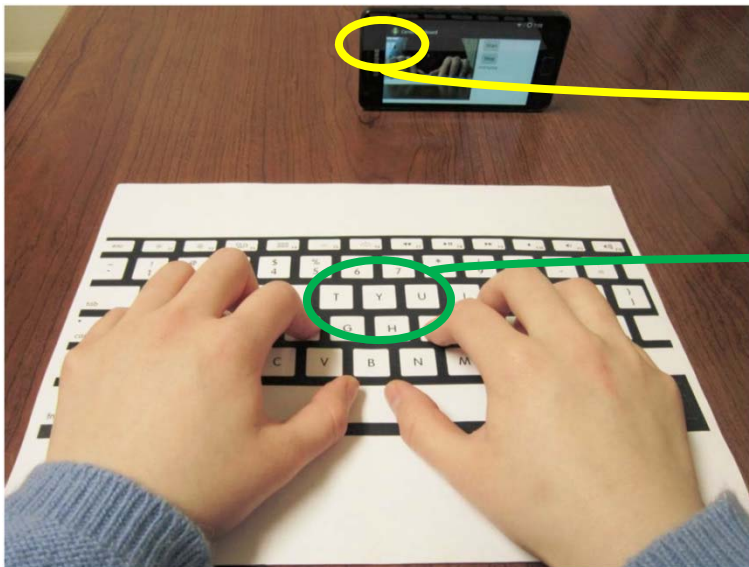


- 移动设备（如智能手机）的前置摄像头；
  - 一张纸键盘；
- 实现一种符合直觉和用户习惯的**文本输入**方式，提供**类似于PC键盘**的输入体验。



# 基于目标定位的微动作感知识别机制

## 挑战



- 仅使用一个摄像头，而不使用麦克风等辅助感知设备；
- 两个按键之间的距离仅为2cm左右。
- 作为一种文本输入方法，需要在可忽略的时延内输出按键对应的字符。

1. **低假阳性率**  
的按键检测



2. **高精度**  
的按键定位



3. **低时延**  
的文本输出

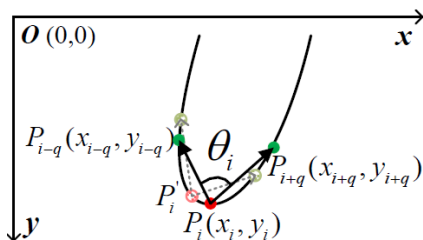
# 挑战1: 按键操作检测

## 指尖检测: 肤色分割、形状检测

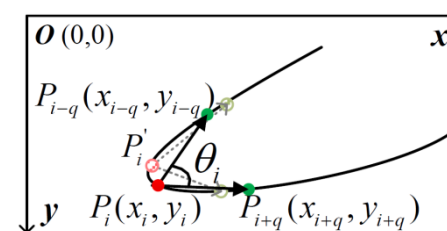
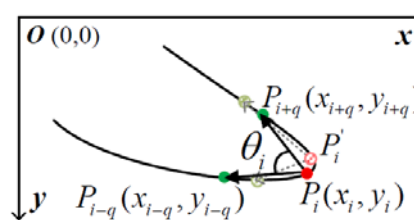
- 集合Ostu's算法和YCrCb空间中的红色通道, 实现肤色分割。
- 采用腐蚀、膨胀操作, 减少指尖周围区域的干扰。
- 利用指尖的形状特征, 检测指尖位置。



### · 非拇指指尖



### · 拇指



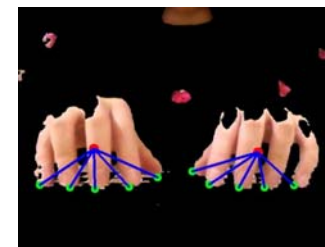
(a) 输入图片



(b) 手部分离



(c) 手部优化

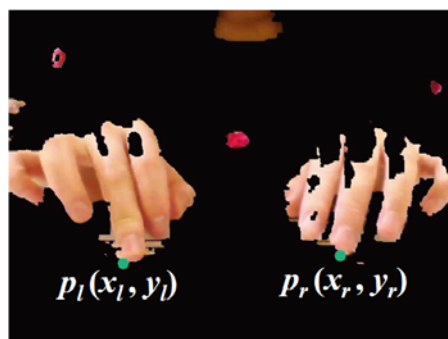


(d) 指尖检测结果

# 挑战1: 按键操作检测

- 指尖跟踪与按键操作检测

1. 追踪StrokeTip, 检测按键操作可能对应的**候选指尖**:



$$P_l(x_l, y_l) \text{ 满足} \\ y_l \geq y_j, l \neq j, P_j \in C_l,$$

$C_l$  表示左手的轮廓点集合.  
(开始时刻, 拇指默认选为候选指尖)

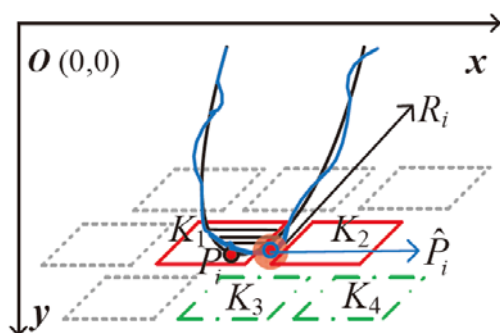
2. 检测**手指运动/停止**, 检测可能的**按键操作**:  
(发出按键操作时, 手指会在该按键上停留一段时间)

$$\sqrt{(x_{l_i} - x_{l_{i-1}})^2 + (y_{l_i} - y_{l_{i-1}})^2} \leq \Delta r, \\ \sqrt{(x_{r_i} - x_{r_{i-1}})^2 + (y_{r_i} - y_{r_{i-1}})^2} \leq \Delta r.$$

## 挑战2: 按键操作定位

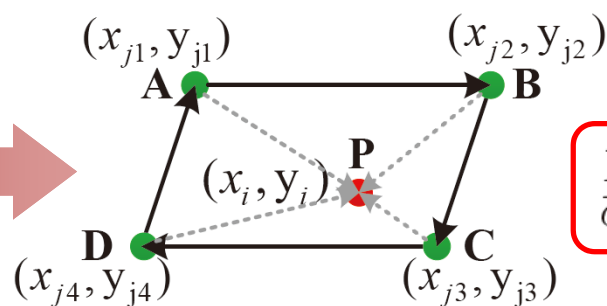
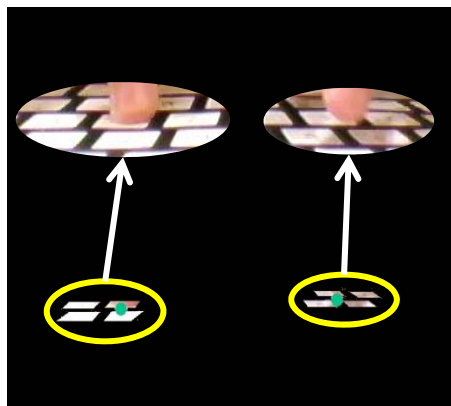
- 匹配按键操作对应的指尖与按键

1. 对每一个候选指尖, 选出其周围的**候选按键**



选取候选按键

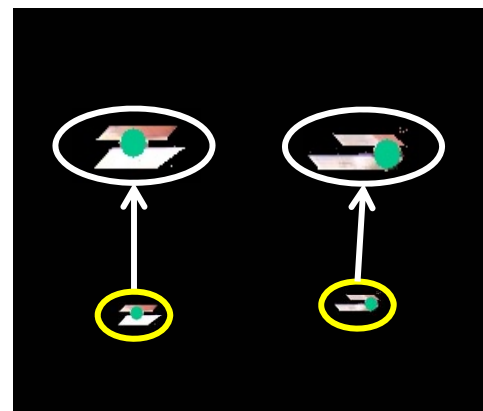
(从最接近的**两行、两列**中选择四个候选按键)



指尖定位

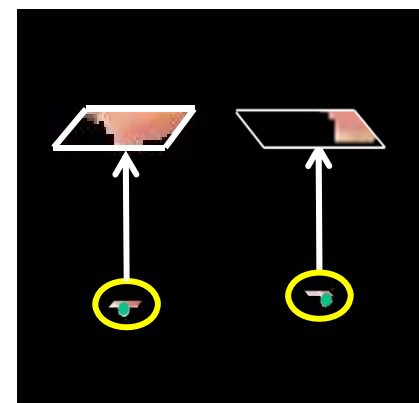
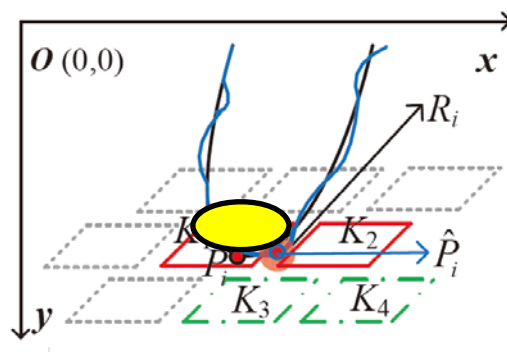
(仅保留包含指尖的候选按键)

$$\begin{aligned} \overrightarrow{AB} \times \overrightarrow{AP} &\geq 0, \overrightarrow{BC} \times \overrightarrow{BP} \geq 0, \\ \overrightarrow{CD} \times \overrightarrow{CP} &\geq 0, \overrightarrow{DA} \times \overrightarrow{DP} \geq 0. \end{aligned}$$

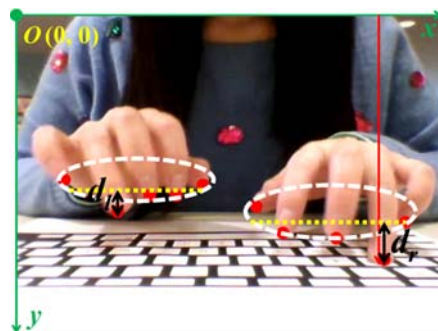


## 挑战2: 按键操作定位

- 匹配按键操作对应的指尖与按键
- 计算**候选按键**在摄像头视野中**被遮挡面积**

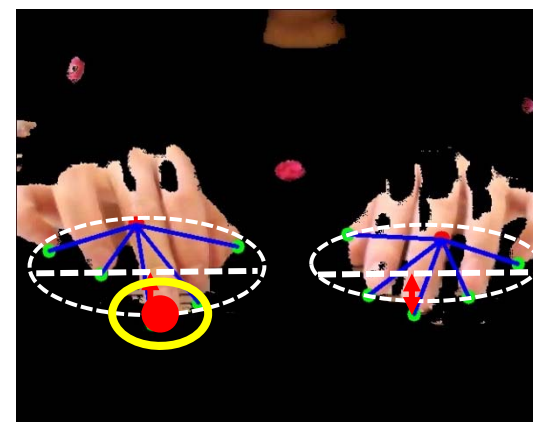


- 计算**候选指尖**与该手上其他指尖的**平均竖直距离**



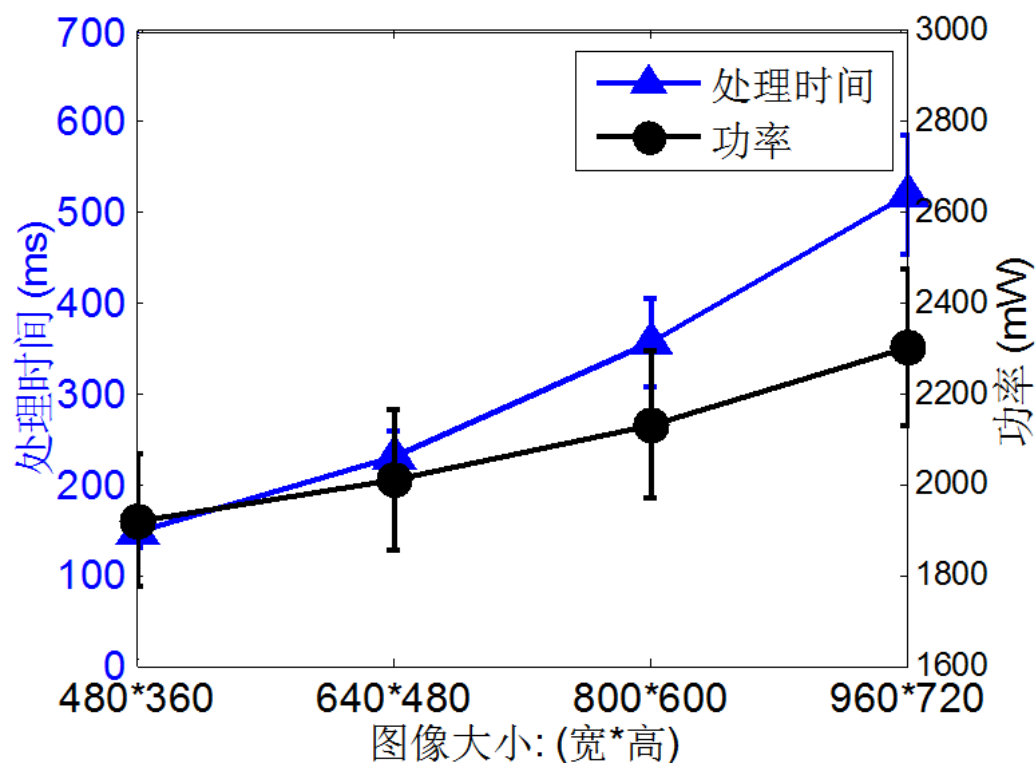
$$d_l = |y_l - \frac{1}{4} \cdot \sum_{j=1}^{j=5} y_j, j \neq l|$$

$$d_r = |y_r - \frac{1}{4} \cdot \sum_{j=6}^{j=10} y_j, j \neq r|$$



## 挑战3: 低时延文本输出

- 计算开销大、时延高



处理一张图片的时间/能耗 ( Samsung GT-I9100 )

➤ **处理一张**640\*480像素点的图片，时延为**630毫秒**；

➤ **捕捉图片**所需的时间，最快为33毫秒获取一帧（帧速率为30fps）；

➤ **按键操作检测**所需的时间，跟踪多张图片检测手指移动/停止状态；



识别一个**按键操作**的时延达**1320毫秒**

## 挑战3: 低时延文本输出

### • 实时图像处理

#### 1. 动态调整图片大小

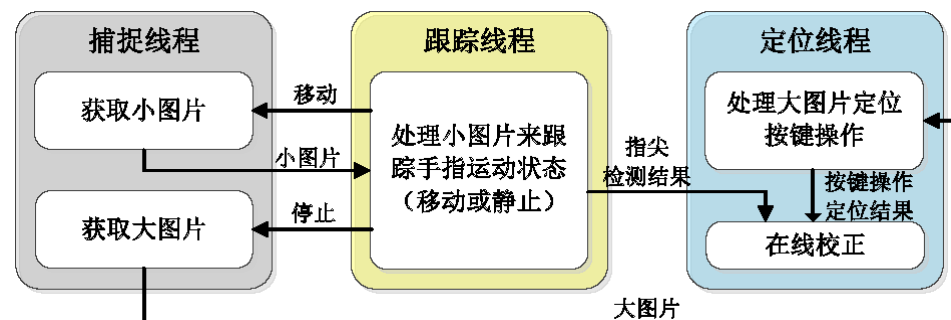


- 两个按键操作之间，采用小图片 (120\*90 pixels)；
- 按键操作定位时采用大图片 (480\*360 pixels)；

#### 2. 聚焦大图片中的目标识别区域

- 仅处理大图片中涉及按键操作的少部分区域，如上图中的红色矩形区域。

#### 3. 引入多线程



- 并行采集图像信息、检测按键操作、定位按键操作。

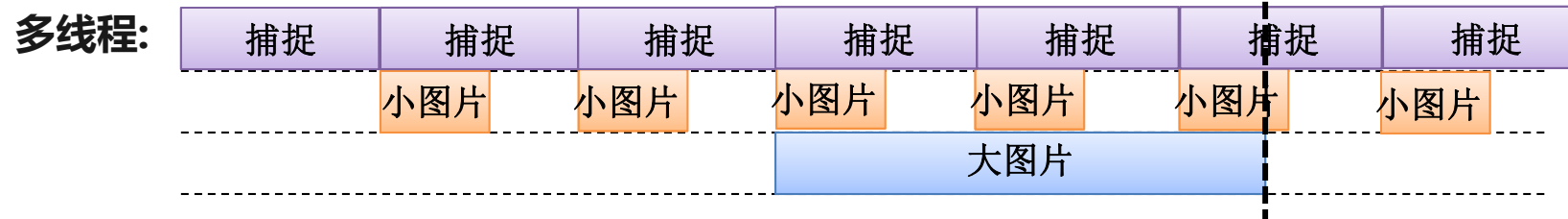
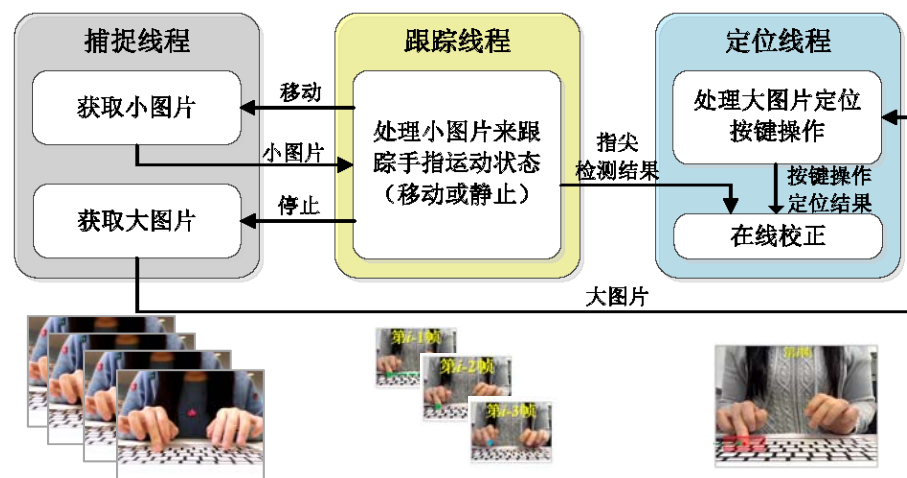
#### 4. 剔除读写图片操作

- 直接以二进制的方式将图像信息存储在文本文件中，剔除频繁读写图片操作。



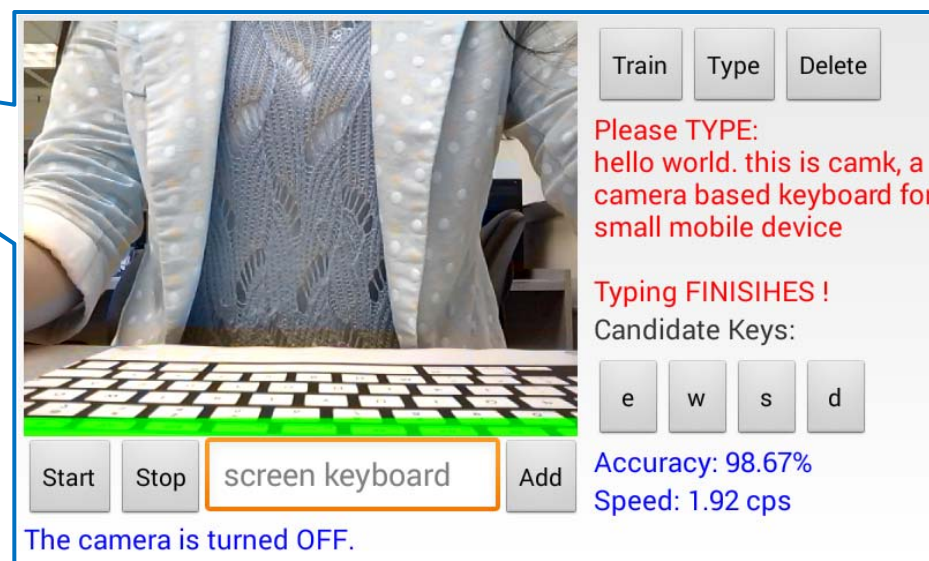
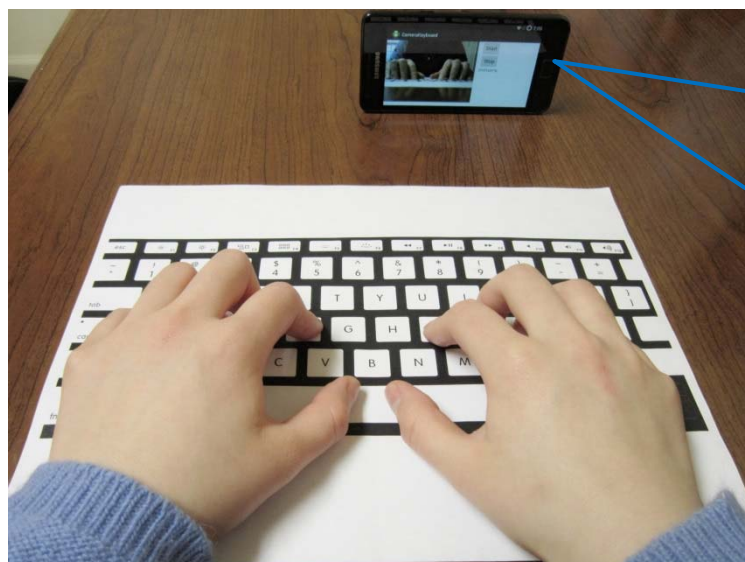
## 挑战3: 低时延文本输出

- 多线程实现并行处理



# 系统实现

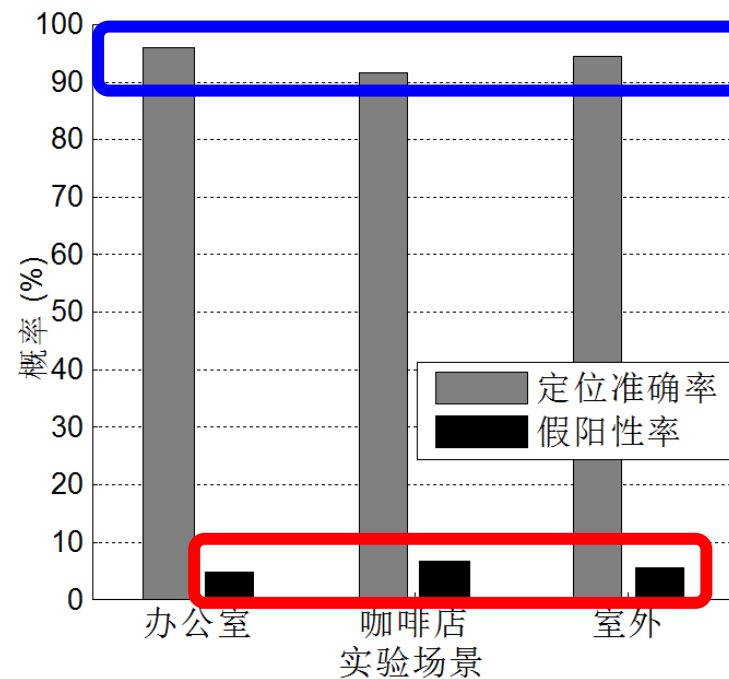
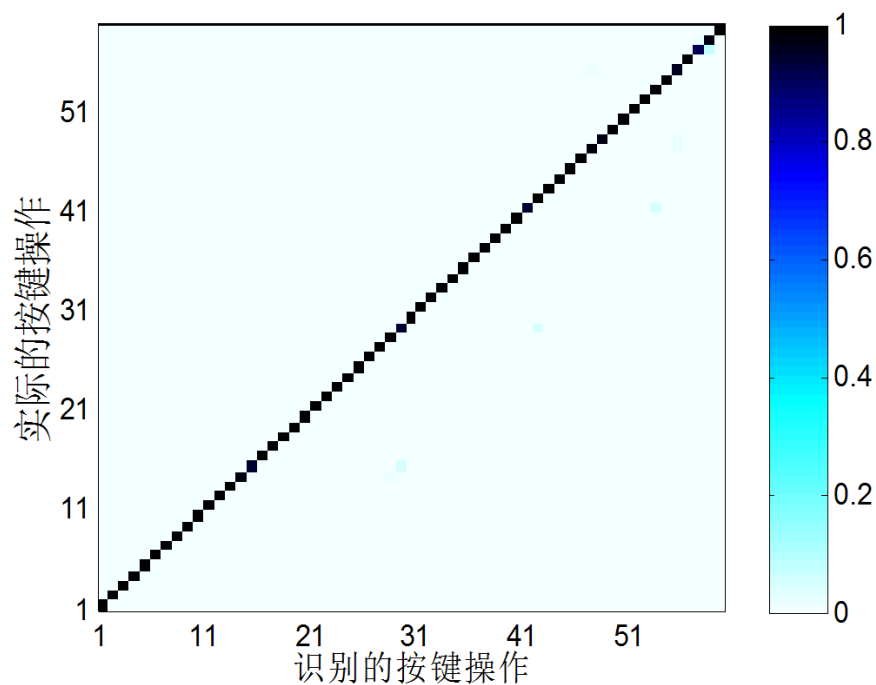
我们在安装Google安卓系统（版本4.4.4）的智能手机上实现了CamK系统。



- **纸键盘:** 美国信纸大小.
- **键盘布局:** 苹果无线键盘布局;
- **智能手机:** 三星 GT-I9100: (CPU: 1228MHz; 内存: 4GB ROM+1GB RAM; 前置摄像头: 2 MP)

# 性能评估

## • 按键操作检测与定位



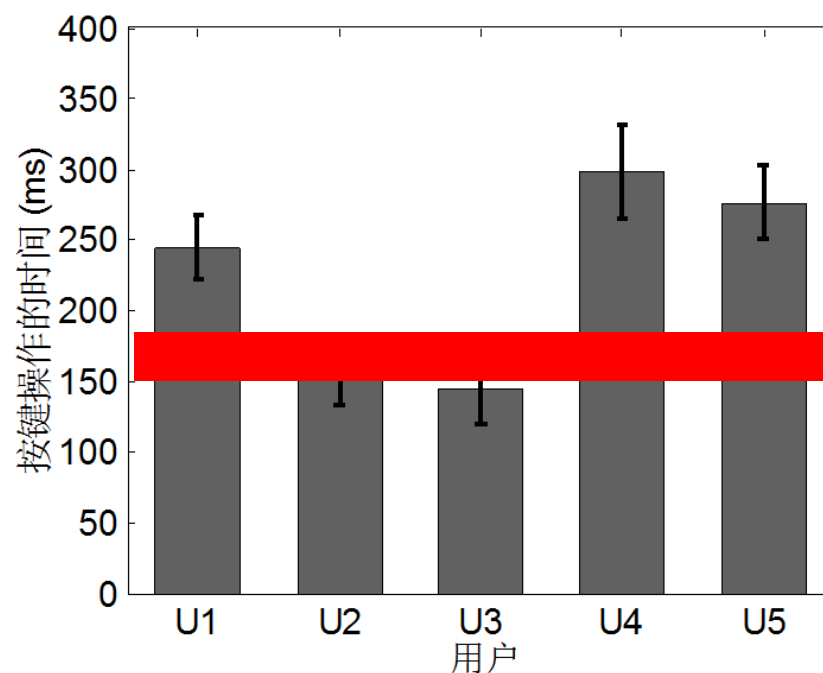
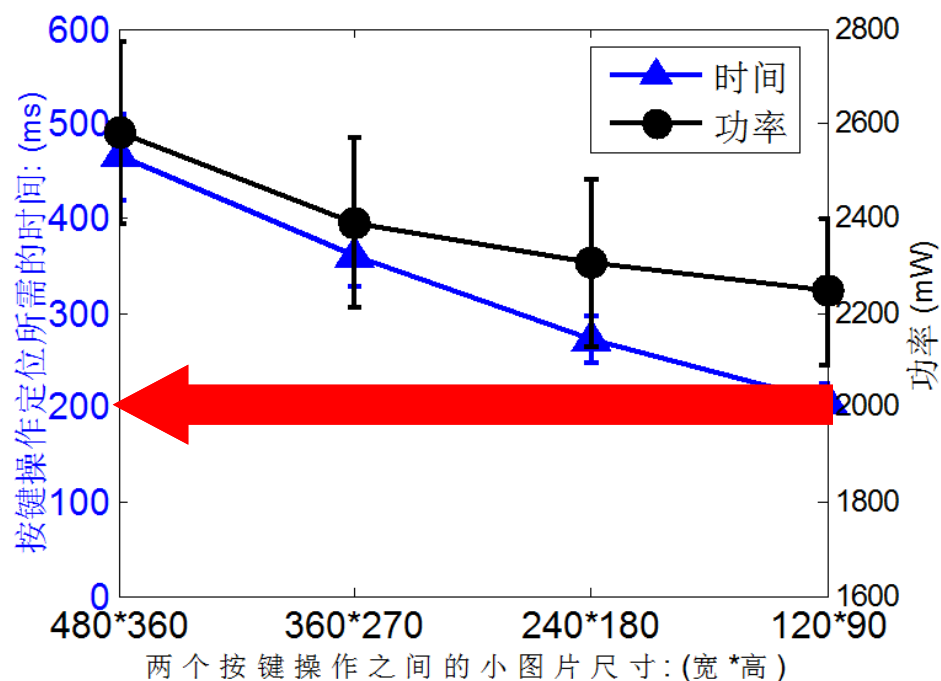
➤ 已知按键操作的定位准确率接近100%；

➤ 低假阳性率：小于5%。

➤ 高识别准确率：大于90%；

# 性能评估

## • 文本输出时延

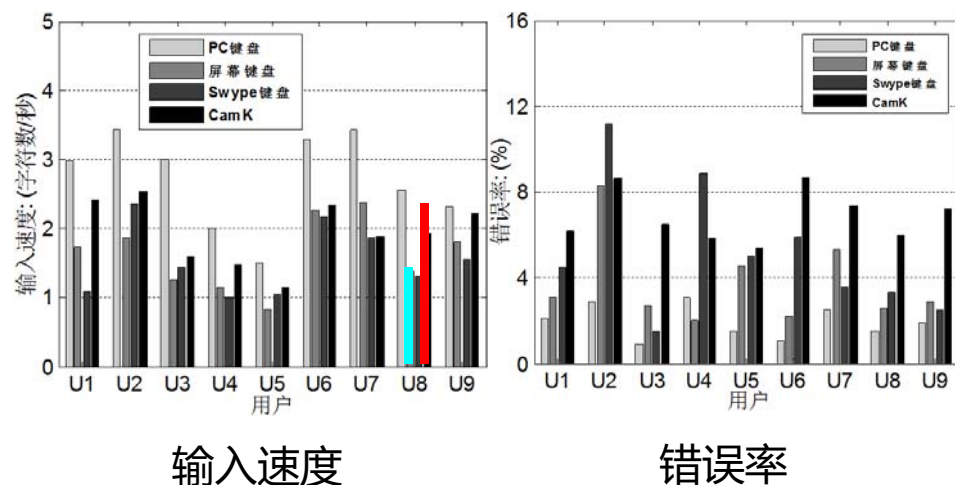


- 检测并定位一个按键操作所需的总时间约为**200ms**，与摄像头中的按键操作时间相当。
- ➔ 按键操作检测与定位的**时延约在50ms之内**，CamK能够以**可忽略的时延**实现文本输出。

# 性能评估

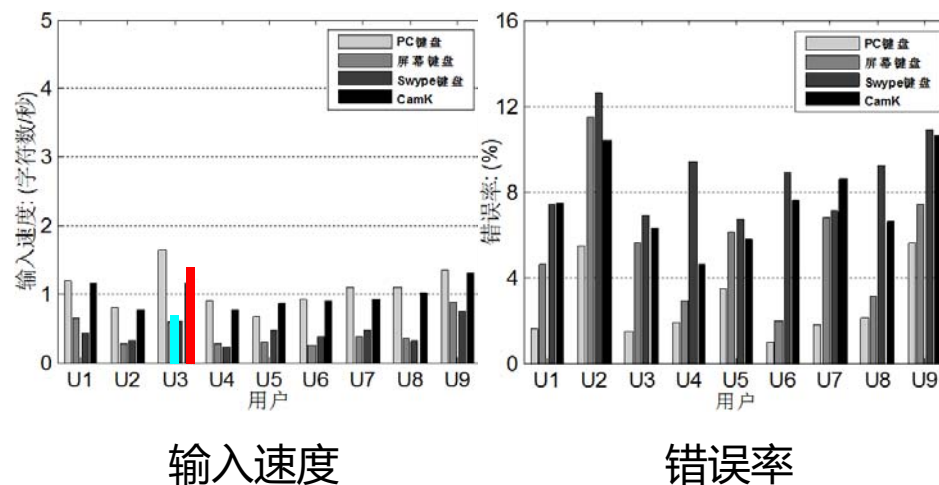
## • 用户使用情况测试

### 常规文本输入



- CamK的文本输入速度是**屏幕键盘的1.25倍**。
- CamK的错误率约为5%~9%，与基于声音信号的UbiK ( 4%~8%，Mobisys'14 ) 相当。

### 随机字符输入



- CamK的字符输入速度是**屏幕键盘的2.5倍**。
- CamK的错误率约为6%~10%，与基于声音信号的UbiK ( 4%~10%，Mobisys'14 ) 相当。

# 总结

- 物联网中的感知识别层：包括RFID系统、无线传感器、移动设备等
- 大规模多阅读器的RFID系统，如何在分布式环境下实现有效的数据感知，避免阅读器-标签冲突和阅读器-阅读器冲突；
- 在计算资源有限的感知终端（如智能手机），如何以并行化的方式减少计算开销、数据存储等；并保证数据的一致性和同步性。



# 谢谢！

